N63-10952

CODE 1

NASA TN D-1678

NASA TN D-1678

TECHNICAL NOTE

D-1678

A LOGICAL NET MECHANIZATION FOR TIME-OPTIMAL REGULATION

By F. B. Smith, Jr.

Prepared Under Contract NASr-27 by
MINNEAPOLIS-HONEYWELL REGULATOR COMPANY
Minneapolis, Minnesota
for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

WASHINGTON December 1962

A LOGICAL NET MECHANIZATION FOR TIME-OPTIMAL REGULATION

By Fred B. Smith, Jr.

SUMMARY

A technique is described for mechanizing relay controllers. A quantized phase space for the plant is mapped into two or more points by a collection of logic elements whose Boolean inputs are the quantized variables. If the switching surfaces are known explicitly, the logic (the mapping function) of the controller may be computed in a straightforward manner. If the surfaces are known only implicitly, the logic may be adjusted on a representative collection of particular values of the desired mapping function.

Experimental results are presented for time-optimal control of third-order and fourth-order plants.

SYMBOLS

x_i^j A Boolean variable representing the jth quanta of the ith variable

= 0 if phase point is not within that quanta

= 1 if phase point is within that quanta

$\cup$ union - logical "or"

$\bigcup_{i=1}^{n} x_i$ $x_1 \cup x_2 \cup x_3 \ldots \cup x_n$

λ threshold element constants

θ pitch attitude

$\dot{\theta}$ pitch rate

α angle of attack

δ gimbal deflection

u control variable

INTRODUCTION

Mechanization of closed-loop "optimal" relay controllers is usually difficult to accomplish satisfactorily. This section of the report provides a means for overcoming two of the basic difficulties: the implementation of the nonlinear control law and the utilization of an open-loop control law for closed-loop control.

The first difficulty arises because a mathematical criterion for optimal quality of control usually demands a nonlinear relationship between the measured state of the system and the control variables. The second difficulty is due to a shortcoming of the theory. Instead of providing an explicit expression for the control u as a function of the state variables x, it most frequently yields the control variable u as a function of time (t) for given initial conditions x(0).

In some cases, these problems can be resolved by repeatedly solving the open-loop control problem on a sufficiently fast time scale compared with the dynamics of the system to obtain an adequate approximation to the desired closed-loop control. This is the method used in ref. 1 to obtain zero-dimensional time-optimal control of a fourth-order system with real roots. Such a procedure provides more information and demands a more complex control computer than is needed for the time-invariant control problem.

It has often been suggested that when the optimum control variable is bang-bang, the switch times of a collection of open-loop solutions could be used to define the closed-loop control law. The method presented here is related to that procedure. A collection of optimum open-loop trajectories is used to define those regions in phase space associated with a positive forcing function and those associated with a negative one. In order than only a finite number of points need be considered, the phase space of measured variables is quantized and a sign associated with each hypercube. This introduces an approximation in the surface separating the regions of different sign. It is assumed that the control can be made acceptably close to optimum by choosing a sufficiently fine quantization. Hopkin and Iwama (ref. 2) took a similar approach in quantizing and assigning signs to the quanta, but suggested a memory location for each quanta using a magnetic drum or core memory. This provides great flexibility but memory requirements become prohibitive. If five variables are each divided into 32 quanta (a reasonable fineness for good approximation to the optimum response), more than 10^7 memory locations are required. It will be shown that it is possible to collect together those points with the same sign in a logic expression and mechanize this with a small number of threshold logic elements without losing the flexibility associated with drum or core storage. Such a procedure is applicable in principle to control of any system in which the control variable, decision, or classification to be made is one of a finite number of states and the desired output state is known for every input state (every possible combination of input variables). When the desired output state is the same for a large number of "neighboring" input states, then only a representative sample of the input and corresponding output state is required. This has provided the basis for many of the pattern recognition experiments and, in the work presented here, forms the basis for obtaining a controller from a relatively small set of representative open-loop optimum trajectories.

CONTROLLER DESIGN

When control specification is given only implicitly through a set of desired responses, it is not possible to completely separate the problems of mechanization from those of controller synthesis. A solution of this combined problem of controller design is suggested in this subsection. It begins with a proposed mechanization of a logic function which expresses an assumed closed-loop control law u(x). If the control law is explicitly known then the problem ends there. If the control law is not known explicitly, a method is provided for adjusting the logic to make the controller fit the known information.

Control Logic

It is first necessary to transform the usual form of closed-loop control law into some form of logic expression. This is done by dividing the region of interest of each of the m variables into k sub-regions called quanta (figure 1a). If

$$r_{j-1} < x_i \leq r_j$$

then the variable x_i is said to be in the jth quanta. A Boolean variable,

$$x_i^j \qquad \begin{array}{l} i = 1, 2 \ldots m \\ j = 1, 2 \ldots k \end{array}$$

is defined for each of the mk quanta. An x_i^j has the value one if the magnitude of the ith measured variable x_i, is in the jth quanta, and has the value zero if the magnitude is not. The m-dimensional phase space is thus divided into k^m hypercubes, each of which is specified by a set of m quanta. A switching function

$$Q = x_1^{j_1} x_2^{j_2} \ldots x_m^{j_m}$$

is defined for each of these hypercubes. This switching function has the value one if a measured phase point is within that hypercube (all $x_i^j = 1$) or the value zero of the phase point is not within it (at least one $x_i^j = 0$). If the number of measurements, m, is equal to the order of the system, n, and the plant being controlled is time-invariant, then control law u(x) will divide the n-dimensional space into regions in which the control variables are uniquely determined. The surface separating these regions is called the switching surface. A unique control variable state (set of unique signs) may be associated with each of the n-dimensional hypercubes except those through which a switching surface passes. These may be assigned a state in any consistent manner. If m < n, then there may be regions in the m space in which the state of the control variable is not uniquely defined.

It is assumed, however, that a state is assigned in some manner to each hypercube. A logic function, F, which is true when the measured phase point is within any of the hypercubes having the same control variable state, may then be written. The function F has the form,

$$F = \cup Q \text{ (all } Q \text{ of same control variable state)} \tag{1}$$

(The $\cup$ notation will be used here for logical "or" to avoid confusion with the algebraic operation of addition to be used later on the Boolean variables.)

For example, the logic function collecting positive hypercubes of figure 1b is,

$$F = \underset{C.V.\oplus}{\cup Q} = x_2^1 \cup x_2^2 \quad (x_1^2 \cup x_1^3 \cup x_1^4) \cup x_2^3 \; x_1^4 \tag{2}$$

A mechanization of this function provides a controller based on the quantized switching surface of figure 1b. Whenever a phase point is on the positive side of the surface, equation (1) is true (one) and vice versa. The logic of equation (2) can be realized by a number of types of hardware, but the type which seems most promising is threshold logic. Any logic function can be represented by a set of threshold elements. Within a given framework of elements, very major changes in the logic function generated can be made by simple changes of constants. The expression for a single threshold logic element is

$$T = \text{sign} \left[\Sigma y_i \; \lambda_i + \lambda_o\right] \tag{3}$$

where

- y_i are Boolean variables having values 0 or 1
- λ_i, $i = 0, 1, 2, \cdots$ are constants
- Σ algebraic summation
- T is considered true ($T = 1$) if the sign of bracket terms is positive or false ($T = 0$) if sign of bracket terms is negative or zero

Quite different logic functions can be mechanized by such an element by simple changes of constants, λ_i. For example, using procedures described in the appendix it is possible to compute a single logic element which correctly maps points on either side of a large class of monotonic surfaces of high dimensions. Although optimum switching surfaces for systems having more than one switch are not monotonic, work presented later in this section of the report shows that the control logic for a number of such switching surfaces can be adequately approximated by a single element. A second-order example shows that some non-monotonic surfaces can be exactly mapped by a single element.

Linear programming provides one technique for computing constants, λ_i, if the logic function is realizable by a single element. However, the set of simultaneous equations and inequalities which must be solved is prohibitively large for this application because of the large number of Boolean variables. A suitable way of finding λ's for the present application is through the use of the constructive theorems of ref. 3; see appendices. These theorems permit complete specification of at least one set of threshold elements for any logic function of the form of equation (1) (Appendix: Corollary 2). This set of elements and their constants is not unique and it is up to the designer to find the most suitable set. As an example of the use of the theorems the threshold element realizing the logic of equation 2 is computed in the appendix. The result is,

$$u = \text{sign}\left[4x_2^1 + 2x_2^2 + x_2^3 + 2x_1^2 + 2x_1^3 + 3x_1^4 - 3\right] \tag{4}$$

It is seen that equation (4) is positive whenever equation (2) is true, and negative when not true and the desired division of space shown in figure 1b is mechanized.

For plants having complex roots the optimum switching surface may not be even approximately monotonic outside the n-1 switching region. For example, the time-optimal switching surface for one-dimensional control of a second-order plant with complex roots is shown in figure 2. In this case it is possible to find a single threshold element to correctly map the points on one side of the quantized surface. Constants for the element are given in table I. It is expected, however, that one element will not, in general, be sufficient for exact mapping of such surfaces for higher order plants.

The quantized switching surface for one-dimensional time-optimal control of a third-order plant with real roots is shown in figure 3. This surface is monotonic in the region of interest and is correctly mapped by the single element

$$u = \text{sign}\left[\sum_{i=1}^{3} \sum_{j=-16}^{+16} x_i^j \lambda_i^j + \lambda_o\right]$$

Constants for this element are given in table II. Because of symmetry about the origin, the surface in only one half of the space is considered. Typical responses for this plant using a controller with this logic are presented later.

Logic Adjustment

The ability to compute threshold elements for mechanizing a given logic expression discussed in the previous paragraphs is of little value for controller mechanization when the switching surface is not known explicitly. For plants of order greater than three, this is the usual case.

Solution of the mathematical optimization problem most often yields a specification for the control variable as a function of time. Each such solution provides implicit information about the switching surface through the switch times. Each open-loop trajectory provides partial information about the regions of positive and negative control variable in the m-space. In the quantized space a finite set of such trajectories, properly distributed throughout the space, is sufficient to define the optimum control variable sign for every hypercube. To obtain such a complete set of trajectories is very time-consuming (expensive) and is in fact, not necessary because of the high probability of a given hypercube having the same sign as its neighbors. If a form or framework of threshold elements is assumed, a trial and adjustment of the logic based on an incomplete set of trajectories can be used to approximate the implicitly defined switching function. Such a procedure is called "training" and is shown schematically in figure 4a. A random initial condition $x(0)$ is chosen from the x-region of interest and the optimal solution $u(t, x(0))$ obtained. This $x(0)$ and $u(t)$ are then applied open-loop to the equations describing the system to be controlled; and at periodic time intervals, the optimum control variable is compared with that specified by the logic network for that $x(t)$. If the two control variables $u(t)$ and $u(x(t))$ are in the same state, no change is made and the equations are integrated through the next time increment. If the two control variables are in different states, the logic of the net is changed by changing the appropriate constants of the threshold elements in a direction tending to give the correct answer for that point. For example, if the logic is of the form,

$$u(x) = \text{sign}\left[\sum_{i=1}^{m} \sum_{j=1}^{k} x_i^j \quad \lambda_i^j\right]$$

and

$$u(t) = +1, \; x(t) = x_1^{h_1}, \; x_2^{h_2} \ldots, \; x_n^{h_n}$$

$$u(x(t)) = -1 \text{ (wrong sign)}$$

then $\lambda_i^{h_i}$, $i = 1, 2 \cdots m$ are each increased by one increment. This makes $\Sigma x_i^j \lambda_i^j$ less negative by m units. If it is still of wrong sign, another correction can be applied or the equations can be integrated through the next time increment and another correction not applied until the next time that point (hypercube) is passed through. Such considerations govern the nature of convergence of the procedure. In most work reported here one correction is applied for each pass through a point.

It is not necessary of course to use open-loop responses in the adjustment. The logic could just as readily be trained to "mimic" another controller operating in a closed loop by placing it in parallel with the controller or by "looking at" operating records.

There are two ways of determining how well and to what extent the logic adjustment is proceeding. The first is to keep track of the percentage of points in error. Typical plots of per cent error versus number of training trajectories are shown in figures 5, 6, and 7 for third- and fourth-order optimal trajectories. It is seen that the number of errors drops very rapidly to less than 10 per cent and then levels off as final adjustments are made on points near the switching surface. The final apparent leveling off is primarily a function of the resolution permitted in the λ_i^j's.

The second method of evaluating the stage of logic adjustment is to use the logic as a controller (figure 4b) to permit qualitative evaluation of the responses. This is a more time-consuming check and is normally done only once or twice during the training to obtain some feel for the control characteristics relative to the training curve.

APPLICATIONS - TIME-OPTIMAL CONTROL

Minimum response time was chosen as the optimization criterion on which to test the ideas previously discussed. The control variable is known to be bang-bang up to the response time, after which it is either turned off (for zero-dimensional control) or a linear control switched in (to control the plant target set about the origin for higher dimensional control). To obtain a representative set of trajectories exterior to the target set it is most efficient to choose a set of initial conditions uniformly distributed in state space and to find the optimum control variable $u(t,x(0)$ by solving the set of transcendental equations described in section 1. Solutions are then selected at random from this set and open-loop trajectories computed at intervals of 0.10 seconds and stored on magnetic tape for the logic adjustment program. The training or logic adjustment program averaged about 20 minutes per thousand training trajectories on a Honeywell H-800 general purpose digital computer. The methods are demonstrated by developing controllers for a typical 250,000-pound launch vehicle at the maximum q flight conditions. Pitch attitude provides fourth-order equations; pitch rate control, third order. Each of these is discussed in the following paragraphs with presentation of simulation results.

One-Dimensional Time-Optimal Control of Third-Order Plant

Rigid vehicle equations are taken to be;

$$\begin{bmatrix} \ddot{\theta} \\ \dot{\alpha} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} -0.0390 & 2.140 & -4.404 \\ 1.0 & -0.0274 & -0.0421 \\ 0 & 0 & -0.02 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \alpha \\ \delta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.2 \end{bmatrix} u \qquad (7)$$

The requirement that pitch rate be brought to zero in minimum time and held there specifies a one-dimensional time-optimal controller. Ideally, $\dot{\theta}$ is brought in minimum time to a line segment in the $\dot{\theta}$, α, δ space and held there with a unique linear control (considered to be the third state of a three-state controller). In this case, the line segment and linear controller are

$$\begin{aligned} \dot{\theta}(T) &= 0 \\ 2.140\,\alpha(T) - 4.404\,\delta(T) &= 0 \\ |\delta(T)| &\leqq 7.18 \end{aligned} \qquad (8)$$

$$u \text{ (after response time)} = -0.1625\,\alpha + 0.1952\,\delta$$

(Illustration and discussion of specification of controllers and target sets are given in sections 1 and 4.)

Because of surface approximation in quantization and non-ideal components, system motion to some small region about the line segment must be accepted. The manner in which the linear control state is switched in is shown in figure 8. The function

$$\left| \sum_{i=1}^{n} \alpha_i x_i \right| - \Delta$$

is used to switch from the bang-bang state to the linear state. The

$$\sum_{i=1}^{n} \alpha_i x_i = 0$$

is the equation of the target line,and when the error in this is less than Δ (chosen to be equivalent to be approximately one quanta) relay R pulls in.

The quantized optimum switching surface for the system is shown in figure 3. This surface representation was obtained by "bracketing" it with solutions for u(t, x(0)) section 1, the x(0)'s being chosen in the center of each cube. The computed constants for collecting with a single threshold element all hypercubes on one side of the surface are given in table II.

Some typical responses of the system with the computed logic are shown in figure 9. Quantization intervals were:

$$\Delta_{\dot{\theta}} = 0.00625 \text{ rad/sec}$$
$$\Delta_{\alpha} = 0.003125 \text{ rad}$$
$$\Delta_{\delta} = 0.00625 \text{ rad}$$

in the range

$$-0.1 \leqq \dot{\theta} \leqq 0.1$$
$$-0.05 \leqq \alpha \leqq 0.05$$
$$0 \leqq \delta \leqq 0.1$$

The effect of quantized switching surface can be noted in the traces of the control variable and gimbal deflection, δ. Pitch rate, $\dot{\theta}$, is brought to within the first quanta in all cases and held there with the theoretical control variable of equation (8), which keeps $\dot{\theta}$ a constant. The necessity for a control function to take out small errors in $\dot{\theta}$ after the response time is shown in the third trace of figure 9. Angle of attack, α, is seen to diverge after the response time because of the small constant error in θ. This is corrected in subsequent simulations by using linear switching to take out residual errors.

To obtain a comparison of computed logic and adjusted logic control, a set of 152 open-loop trajectories with initial conditions fairly uniformly distributed throughout the space of interest were obtained. Trajectories with switches outside the region of interest were omitted even though the initial conditions were within. This set of trajectories was taken in random order and used to adjust a single threshold element in the manner shown in figure 4a and discussed earlier. The per cent error, (errors in N points)/N as a function of the number of trajectories used, is shown in the training curve of figure 5. There was considerable scatter in the individual points and only the smooth curve trend is shown. First switch points are those between the initial condition and the first switch time, second switch points those between the first switch time and the second, etc. Initial constants for the logic were all zero. After repeating these 152 trajectories 18 times in the training procedure, the weights of Table III and the closed-loop responses of figure 10 were obtained. These responses are good approximations to those of figure 9 using computed logic. Linear switching bringing $\dot{\theta}$ and $\ddot{\theta}$ to zero was used in the third control variable state instead of the control of equation (8). The trend of the training curve is still downward, and further training would undoubtedly improve the approximation to optimum response.

One-Dimensional Time-Optimal Control of a Fourth-Order Plant

The fourth-order plant equations are obtained by adding an equation for pitch attitude to equation(7). The plant equations become,

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\alpha} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.0395 & 2.140 & -4.404 \\ 0 & 1.0 & -0.0274 & -0.0421 \\ 0 & 0 & 0 & -0.02 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \alpha \\ \delta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.2 \end{bmatrix} u \qquad (9)$$

Requiring pitch attitude be brought to zero in minimum time and held there specifies a one-dimensional target,

$$\begin{aligned} \theta(T) &= 0 \\ \dot{\theta}(T) &= 0 \\ 2.140\,\alpha\,(T)t - 4.404\,\delta(T) &= 0 \\ |\delta(T)| &\leq 7.18 \\ u\,(\text{after } T) &= -0.1625\,\alpha + 0.1952\,\delta \end{aligned} \qquad (10)$$

It was not practical to bracket the four-dimensional surface as was done in the third-order sample. A set of approximately 400 initial conditions was chosen at uniform intervals in the arbitrarily defined region of interest of the four space, and the method of section 1 was then used to obtain the optimum forcing function for each initial condition. These solutions were taken in random order, points on open-loop trajectories computed at intervals of 0.10 seconds, and the quantized coordinates of each point stored on magnetic tape. Trajectories with one or more switch points outside the region of interest were omitted. This provided a set of 198 trajectories (2435 points) in the space,

$$\begin{aligned} 0 &\leq \theta \leq 0.1 \\ -0.12 &\leq \dot{\theta} \leq 0.12 \\ -0.1 &\leq \alpha \leq 0.1 \\ -0.12 &\leq \delta \leq 0.12 \end{aligned} \qquad (11)$$

Quantization intervals in this space were,

$$\begin{aligned} \Delta_{\theta} &= 0.00625 \\ \Delta_{\dot{\theta}} &= 0.00750 \\ \Delta_{\alpha} &= 0.00625 \\ \Delta_{\delta} &= 0.00750 \end{aligned} \qquad (12)$$

for a total of 2^{19} = 524,288 four-dimensional cubes. The sample set of trajectories was then used to adjust the logic of single threshold element in the manner previously described. The training curve of figure 6 shows the convergence of the adjustment procedure. Initial logic constants were taken to be zero. As in the training of the third-order logic, the errors are down to 10 per cent after only 100 trajectories. After 1000 trajectories, errors are down to approximately five per cent, and after 5000 trajectories down to about 2.5 per cent. The ultimate fractional errors attainable are primarily determined by the functional form assumed for the logic (in figure 6, functional form is a single threshold element) and by the resolution permitted in the logic constants. Resolution is defined here as the ratio of the maximum λ_i^j to the increment of λ_i^j when an error is made. The resolution is increasing during the training process (figure 6) and at 5000 trajectories is artifically increased by a factor of two. This causes a pronounced jump downward in the fractional errors, especially second and third switch points. At 7500 trajectories, it is again increased by two, making it approximately one part in 80. At 11,000 trajectories, it is increased to approximately one part in 800. Closed-loop control responses using logic at three different stages of training (198, 2100, and 11,000 trajectories) are shown in figure 11. Resolution of one part in 80 was the limit of the available hardware for the fourth-order controller so that control could not be evaluated at 13,000 trajectories. The plant is statically unstable. It is seen that the first controller has not yet stabilized it although the errors are less than 10 per cent. The second controller (figure 11b) has apparently stabilized the system but does not provide anything approaching optimum response. The third and final controller (figure 11c), after 11,000 trajectories does provide a good approximation to an optimum controller.* Additional typical responses for the final controller are given in figures 12 and 13. As in the third-order system, a linear switching is used as the third state of the control variable and reduces the small residual errors within the smallest quanta. The switching function used is given in figure 8.

The same sample of trajectories was used to adjust the logic of the next step up in logical complexity - a set of orthogonal threshold elements. (Orthogonality meaning that one and only one element of the set will have an output for any input.) Such a set was obtained by dividing the phase space into non-intersecting regions and assigning a threshold element to each one. The training curve for a set of 16 elements is shown in figure 7. The 16 regions in the space of equation (11) were defined by the hyperplanes,

$$\begin{aligned} \theta &= 0.05 \\ \dot{\theta} &= 0 \\ \alpha &= 0 \\ \delta &= 0 \end{aligned} \tag{13}$$

*Logic constants for the controller are given in table IV.

and the boundaries of the region of interest, equation (11). It is seen that convergence is somewhat more rapid than in the single element case. Equivalent error levels were reached with a resolution in λ's of approximately one-half that required for the single element. Limitations of time and equipment prevented analog investigation of this system.

Mechanization

The threshold element required for mechanization of the third- and fourth-order controllers described here is of unusual nature. It has a large number of inputs (112 for the fourth-order system) only n of which are "active" for any given input. Its weighting elements must have quite high resolution (approximately one part in 400 for the computed third-order logic). To accomplish this most rapidly and economically for the simulation study, the outputs of commercial ladder networks, one for each of the n variables, were summed in a standard analog amplifier. Quantization of the analog variables was accomplished by a standard analog-to-digital conversion channel for each variable. A binary number of four bits plus sign (quantizing magnitude to one part in 16) was found adequate for the systems investigated. Diode-transistor logic was used to decode the binary number and apply the proper inputs to the ladder network. The system is shown schematically in figure 14. To obtain the resolution required from the six bit ladder (one part in 64), it was sometimes necessary to subtract a linear function of the coordinates x_i from the discrete switching functions. The linear function was chosen so that the deviations from the line at the center of each quanta x_i^j was a multiple of the smallest ladder step and the maximum deviation was less than 64. In these cases the actual switching function mechanized was

$$\operatorname{sign}\left[\sum_{i=1}^{n}\sum_{j=-16}^{+16} x_i^j \lambda_i^j + \sum_{i=1}^{n} \alpha_i x_i\right]$$

Discussion of Simulation

A number of points in the simulation results indicate several conclusions that are worthy of special note. The quantization of the state variable magnitude to one part in 16 (four bit plus sign binary numbers) appears to be adequately fine for switching surface quantization if a linear switching is used to reduce residual errors within the smallest quanta. A nonlinear quantization with greater accuracy near the origin or target set might be advantageous. A short investigation of this proved inconclusive.

A surprisingly small set of representative trajectories defines a good controller through the logic training procedure. In the fourth-order example, the set of 198 trajectories with points in less than 0.5 per cent of the hypercubes in the phase space gave a good approximation to an optimum controller. The effectiveness of these points is greater than a set of randomly selected points because after the first switch every point is on the switching surface. The set of initial conditions on which the controller was evaluated was much larger than the typical ones presented in figures 12 and 13. In addition to evaluation on the fourth-order plant, the logic was used as the controller for a 13th-order flexible launch vehicle (section 1). Responses for all initial conditions were good. Departures from optimum are primarily attributed to either the small sample size or the incorrect classification by the logic of the points being trained on. Although less than two per cent of these points are in error at the stage of the analog test, those are the critical points on or near the switching surface. The resolution of the logic (one part in 80) at this stage is a limiting factor in the correct classification of points. When the resolution is increased by 10 (to one part in 800), errors drop to about one per cent.

CONCLUSIONS

It is possible to store information about surfaces or regions in phase space of high dimension without complex hardware by quantizing coordinates and collecting members of the same sets with logic elements.

Quantization may be relatively coarse and still give good control.

If limited information about the space is known through a number of sample responses with desired characteristics, a controller may be obtained by adjusting logic to give these responses. If the sample set is sufficiently representative, then the controller obtained will provide control with desirable characteristics for a much wider class of inputs than the sample set.

Using the time-optimal criterion to obtain a set of sample responses, controllers for plants up to fourth-order with real roots may be designed quickly (a few hours computer time) by simulating the system and adjusting the logic digitally.

The design procedure seems practical and applicable to plants of higher order than fourth and to plants with complex roots.

Physical implementation of fixed controllers is straightforward. The resulting controller is simple and could be made more reliable than a digital computer. Implementation is well suited to microminiaturization or molecular deposition techniques.

Additional work is desirable in the area of specifying a more general set of threshold logic elements than those investigated here. Such a set should increase the speed of convergence of the adjustment procedure, reduce the resolution required in the weighting elements to a low level, and be capable of mapping more complex surfaces. A multi-level set of threshold elements capable of mechanizing logic collecting points on one side of any continuous surface seems a distinct possibility. This work should take into consideration the nature of elements best suited for physical implementation (number of inputs per element, resolutions easily attainable, etc.).

Additional work is also desirable in the area of utilizing the adjustable nature of the logic. Such a study would investigate criteria for adjustment based on the performance of the logic itself (such as smoothness or continuity of output). or performance of the plant under control by it or a combination of the two. The study should also investigate the various "variable memory" devices available, possibly suggest additional ones,and construct a logic system for simulation purposes based on the most promising.

This investigation was conducted at the Minneapolis-Honeywell Regulator Company under the sponsorship and with the financial assistance of the National Aeronautics and Space Administration.

Minneapolis-Honeywell Regulator Company
Minneapolis, Minnesota
July 31, 1962

APPENDIX

COMPUTATION OF THRESHOLD ELEMENTS

Four theorems of ref. 3 were useful in the work reported. Slightly modified forms of these theorems are given here. Two corollaries applicable to division of n-dimensional space by threshold elements are proved. An example computation of a threshold element for a switching surface is given.

Definition: The threshold operator, T, converts the algebraic expression $g = a_1 x_1 + a_2 x_2 + \ldots a_n y_n - a_o$, into a switching function, F, having the value zero or one depending on whether g is greater than zero, i. e.,

$$F = T(g) = 1 \qquad g > 0$$
$$= 0 \qquad g \leq 0$$

Physical implementation of T(g) is called a threshold element.

Theorem I: If switching function F is of the form $F = \prod_{i=1}^{n} f_i$, where f_i are logic functions having value 0, 1, then an algebraic function g such that $F = T(g)$ is,

$$g = \sum_{i=1}^{n} f_i - (n-1)$$

Corollary 1: Each hypercube in n-dimensional phase space is specifiable by such an algebraic expression. As discussed in the text, the switching function, Q, representing each hypercube is specified by an expression of the form,

$$Q = x_1^{j_1} \; x_2^{j_2} \ldots, x_n^{j_n}$$

where x_i^j take on values 0, 1 depending on whether a phase point in question is within that quanta or not. Consequently from theorem I

$$Q = T(g) = T\left[\sum_{i=1}^{n} x_i^j - (n-1)\right]$$

Theorem II: If a switching function F is of the form $F = f_1 \cup f_2 \ldots \cup f_n$, then an algebraic function g such that $F = T(g)$ is

$$g = \sum_{i=1}^{n} f_i$$

Corollary 2: A switching function U, collecting any set of hypercubes in an n-dimensional space is representable by a collection of threshold elements $U = T\left[\Sigma T(g)\right]$. This follows directly from corollary 1 and theorem II.

Definition: Two algebraic functions, g_1 and g_2, are said to be equivalent if

$$\text{whenever } g_1 > 0, \quad g_2 > 0$$
$$g_1 \leq 0, \quad g_2 \leq 0$$
$$\text{and when } g_2 > 0, \quad g_1 > 0$$
$$g_2 \leq 0, \quad g_1 \leq 0$$

Definition: a) min $g > 0$ is the smallest value greater than zero taken by g when the Boolean variables of g take on all possible combinations of 0, 1. b) Max $g \leq 0$ is the largest value less than or equal to zero taken on by g under such conditions. c) Min $g\ (x_i = 1) > 0$, Max $g\ (x_i = 1) \leq 0$ are Min $g > 0$ and Max $g \leq 0$ under conditions that x_i is considered constant equal to 1.

Theorem III: Let $g = \sum_{i=1}^{n} a_i Y_i - k, \ k \geq 0$

a) If Min $g > 0$ is p, $g_1 = g - s$ is equivalent to g, $0 \leq s < p$.

b) If Max $g \leq 0$ is p, $g_1 = g + r$ is equivalent to g, $0 \leq r \leq p$.

c) If Min $g\ (x_j = 1) > 0$ is p, g which is obtained by changing a_j to $a_j - s$ is equivalent to g, $0 \leq s < p$, $1 \leq j \leq n$.

d) If Max $g\ (x_j = 1) \leq 0$ is $-p$, g, obtained by changing a_j to $a_j + r$ is equivalent to g, $0 \leq r \leq j$, $1 \leq j \leq n$.

e) Functions g and Kg are equivalent for any real $K > 0$.

Theorem IV: A switching function of the form $F_1 = T\left[T(g_1) + T(g_2)\right]$ is equivalent to the switching function $F_2 = T(Kg_1 + g_2)$, where $K > (k_1/p)$, p is Min $g_1 > 0$, if the following two conditions are satisfied:

a) Whenever $g_2 > 0$, $g_1 \geq 0$

b) Whenever $g_1 > 0$, $g_2 + k_1 \geq 0$

As an example of the use of these theorems, the threshold logic element collecting points on the positive side of the quantized surface of figure 1 is computed. The logic function collecting the squares is

$$F = x_1^2 \cup x_2^2 \quad (x_1^2 \cup x_1^3 \cup x_1^4) \cup x_2^3 \quad x_1^4$$

Define additional switching functions

$$f_1 = x_2^1$$

$$f_2 = x_2^2 (x_1^2 \cup x_1^3 \cup x_1^4) = x_2^2 \; h_2$$

$$f_3 = x_2^3 \; x_1^4$$

By theorem I, the threshold element which is true only when f_1 is true is

$$T(g_1) = T(x_2^1)$$

Similarly, the threshold element which is true only when f_2 is true is

$$T(g_2) = T(x_2^2 \, h_2) = T(x_2^2 + h_2 - 1)$$

and the threshold element which is true only when h_2 is true is, by theorem II.

$$h_2 = T(x_1^2 + x_1^3 + x_1^4)$$

Consequently,

$$T(g_2) = T\left[T(x_2^2) + T(x_1^2 + x_1^3 + x_1^4) - 1\right]$$

But by theorem IV the threshold element equivalent to this is

$$T\left[x_2^2 + x_1^2 + x_1^3 + x_1^4 - 1\right]$$

because Min $x_2^2 > 0 = 1$ and whenever $(x_1^2 + x_1^3 + x_1^4) > 0$, $x_2^2 \geq 0$, whenever $x_2^2 > 0$, $(x_1^2 + x_1^3 + x_1^4) \geq 0$, i. e., $k_1 = 0$ and $\therefore$ choose $K = 1 > \frac{0}{1}$.

The threshold element which is true only when f_1 or f_2 is true is by heorem II and IV.

$$\begin{aligned} f_1 \cup f_2 &= T\left[T(x_2^1) + T(x_2^2 + x_1^2 + x_1^3 + x_1^4 - 1)\right] \\ &= T\left[2x_2^1 + x_2^2 + x_1^2 + x_1^3 + x_1^4 - 1\right] \end{aligned}$$

because Min $x_2^1 > 0 = 1$, and whenever $x_2^1 > 0$, $x_2^2 + x_1^2 + x_1^3 + x_1^4 \geq 0$, i. e., $k_1 = 1$ hence $K = 2 > \frac{1}{1}$.

The threshold element which is true only when f_3 is true is

$$f_3 = T(x_2^3 + x_1^4 - 1)$$

By theorems II and IV

$$f_1 \cup f_2 \cup f_3 = T\left[T(g_1) + T(g_2)\right]$$

$$T\left[T(2x_2^1 + x_2^2 + x_1^2 + x_1^3 + x_1^4 - 1) + T(x_2^3 + x_1^4 - 1\right]$$

$$= \quad T\left[4x_2^1 + 2x_2^2 + x_2^3 + 2x_1^2 + 2x_1^3 + 3x_1^4 - 3\right]$$

because Min $g_1 > 0 = 1$, and when $g_2 > 0$, $g_1 = 0$, when $g_1 > 0$, $g_2 + 1 \geq 0$, i.e., $k_1 = 1$, hence, $K = 2 > \frac{1}{1}$.

REFERENCES

1. Smith, Fred B., Jr.: Time-Optimal Control of Higher Order Systems, IRE Trans. AC-6, Number 1, February 1961, pp. 16-21.

2. Hopkin, A. M. and Iwama, M.: A Study of a Digitally Programmed Optimum Relay Servomechanism for Non-Linear Control of an Airframe, Series No. 60, Issue No. 177, Institute of Engineering Research, University of California, Berkeley, California, February 18, 1957.

3. Wataru Mayeda: Synthesis of Threshold Networks by Alogic Functions, Report R-124, C.S.L., University of Illinois, March 1961.

Table I. - Logic element constants

	λ_1^j	λ_2^j
j = 1	1	0
2	2	0
3	3	0
4	4	0
5	5	0
6	5	-1
7	5	-4
8	6	-3
9	7	-2
10	8	-1
11	9	-5
12	9	-5
13	9	-7
14	9	-8
15	9	-7
16	9	-6

$$u = \operatorname{sign}\left[\sum_{i=1}^{2} \sum_{j=1}^{16} x_i^j \lambda_i^j\right]$$

Computed logic element constants for the quantized surface of figure 2.

One-dimensional control of the plant

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u$$

Table II. - Logic element constants

	λ_1^j	λ_2^j	λ_3^j
j = 16	310	84	- 1
15	295	78	- 26
14	283	72	- 51
13	267	66	- 73
12	254	60	- 96
11	239	54	-122
10	223	52	-145
9	206	42	-166
8	190	40	-189
7	172	34	-212
6	152	28	-235
5	130	26	-259
4	107	20	-284
3	81	9	-306
2	51	5	-330
1	+ 2	2	-353
- 1	-122	- 4	-
- 2	-144	-12	-
- 3	-144	-16	-
- 4	-144	-22	-
- 5	-144	-28	-
- 6	-144	-35	-
- 7	-144	-39	-
- 8	-144	-46	-
- 9	-144	-52	-
-10	-144	-58	-
-11	-144	-61	-
-12	-144	-66	-
-13	-144	-76	-
-14	-144	-80	-
-15	-144	-84	-
-16	-144	-90	-

$\lambda_o = 55$

$$u = \text{sign}\left[\sum_{i=1}^{3} \sum_{j=-16}^{+16} x_i^j \lambda_i^j + \lambda_o\right]$$

Computed logic element constants for the surface of figure 3.

One-dimensional control of the plant

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -0.0394 & 2.140 & -4.404 \\ 1.00 & -0.0274 & -0.0421 \\ 0 & 0 & -0.02 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.2 \end{bmatrix} u$$

Table III. - Logic element constants

	λ_1^j	λ_2^j	λ_3^j
j = 16	69	32	- 59
15	63	29	- 56
14	57	30	- 47
13	45	33	- 40
12	42	23	- 32
11	34	25	- 29
10	28	27	- 22
9	22	18	- 12
8	20	17	- 10
7	16	16	- 2
6	7	17	0
5	2	12	15
4	- 7	12	16
3	-13	10	14
2	-24	9	33
1	-48	9	40
- 1	-	7	39
- 2	-	4	52
- 3	-	5	48
- 4	-	6	61
- 5	-	5	67
- 6	-	- 5	73
- 7	-	- 4	79
- 8	-	- 9	85
- 9	-	- 8	91
-10	-	- 3	97
-11	-	-11	103
-12	-	- 8	109
-13	-	-15	115
-14	-	-16	121
-15	-	-13	127
-16	-	-15	133

$$u = \text{sign}\left[\sum_{i=1}^{3} \sum_{j=-16}^{+16} x_i^j \lambda_i^j\right]$$

Adjusted logic element constants for plant of figure 3.

Set of 152 trajectories repeated 18 times.

One-dimensional control of the plant

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -0.0394 & 2.140 & -4.404 \\ 1.00 & -0.0274 & -0.0421 \\ 0 & 0 & -0.02 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.2 \end{bmatrix} u$$

Table IV. - Logic element constants

	λ_1^j	λ_2^j	λ_3^j	λ_4^j
j = 16	45	52	34	-70
15	46	70	32	-60
14	40	70	34	-50
13	36	70	30	-52
12	33	72	26	-46
11	27	77	30	-34
10	24	70	22	-39
9	24	46	25	-45
8	22	63	17	-13
7	16	65	15	-21
6	13	56	20	- 5
5	6	46	13	-11
4	0	47	10	- 6
3	- 9	32	11	+ 7
2	-26	20	7	4
1	-35	14	6	12
- 1	-	17	7	14
- 2	-	6	5	19
- 3	-	- 1	- 7	24
- 4	-	- 8	2	26
- 5	-	- 9	1	28
- 6	-	-13	- 6	33
- 7	-	-25	- 7	37
- 8	-	-26	- 5	39
- 9	-	-33	- 5	45
-10	-	-40	- 4	50
-11	-	-39	-10	57
-12	-	-43	-11	58
-13	-	-51	-12	58
-14	-	-47	-11	58
-15	-	-49	-15	56
-16	-	-51	-11	69

$$u = \text{sign}\left[\sum_{i=1}^{4} \sum_{j=-16}^{+16} x_i^j \lambda_i^j\right]$$

Adjusted logic element constants for one-dimensional control of the plant

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.0394 & 2.140 & -4.404 \\ 0 & 1.00 & -0.0274 & -0.0421 \\ 0 & 0 & 0 & 0.02 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.2 \end{bmatrix} u$$

Set of 198 trajectories repeated 18 times.

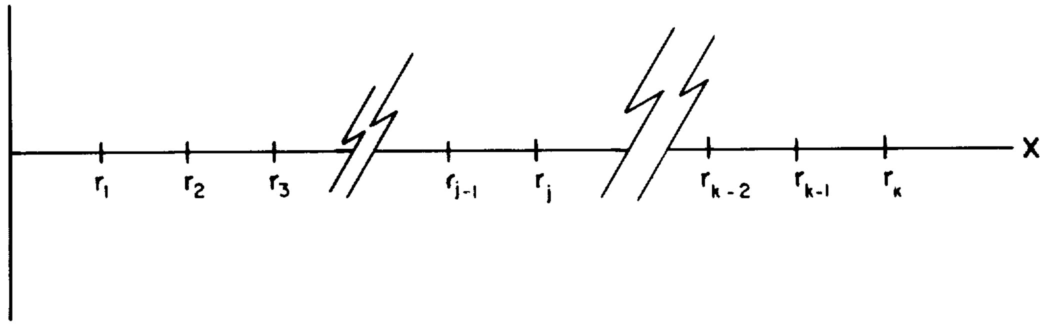


Figure 1(a). - Quantization

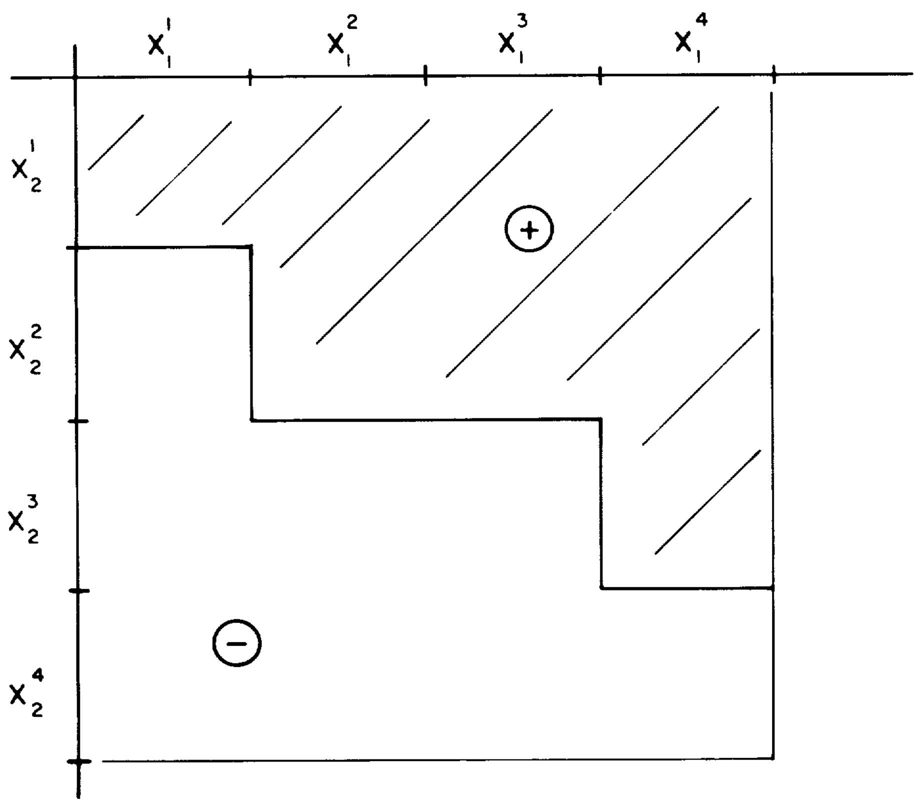


Figure 1(b). - Simple quantized switching surface

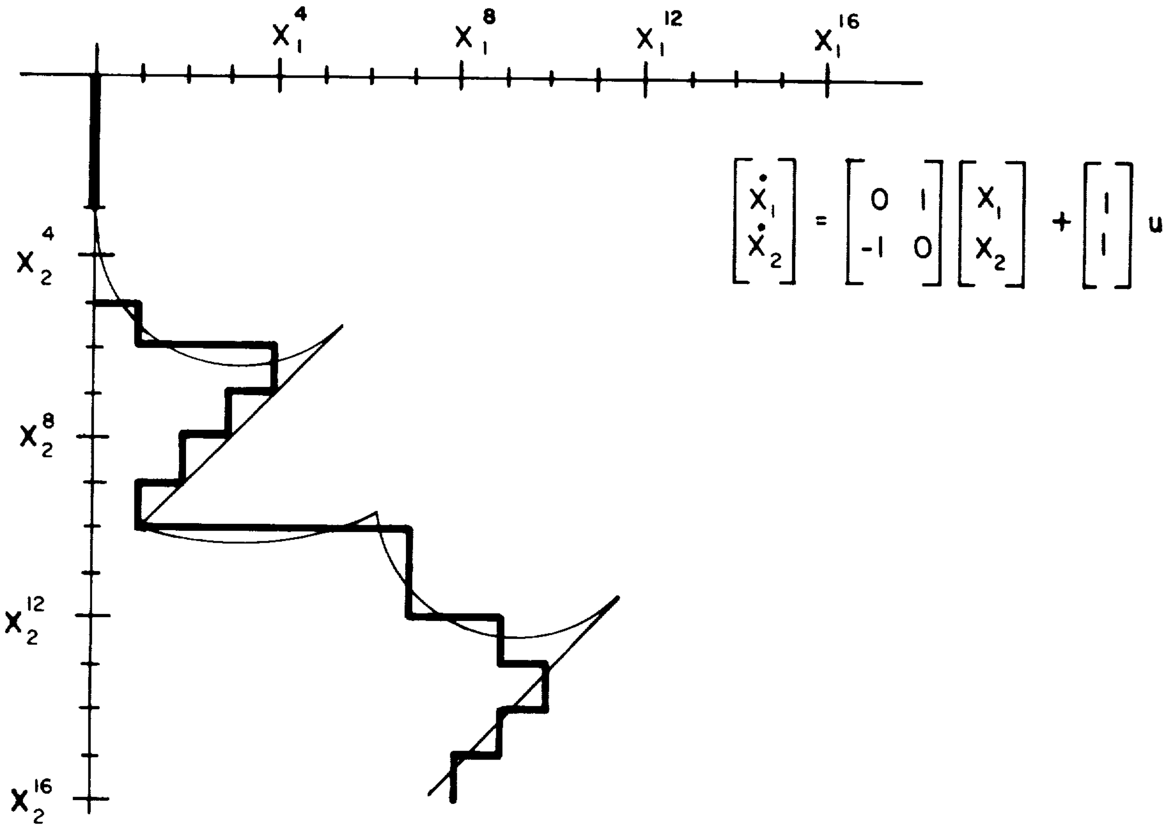


Figure 2. - Complex quantized switching surface

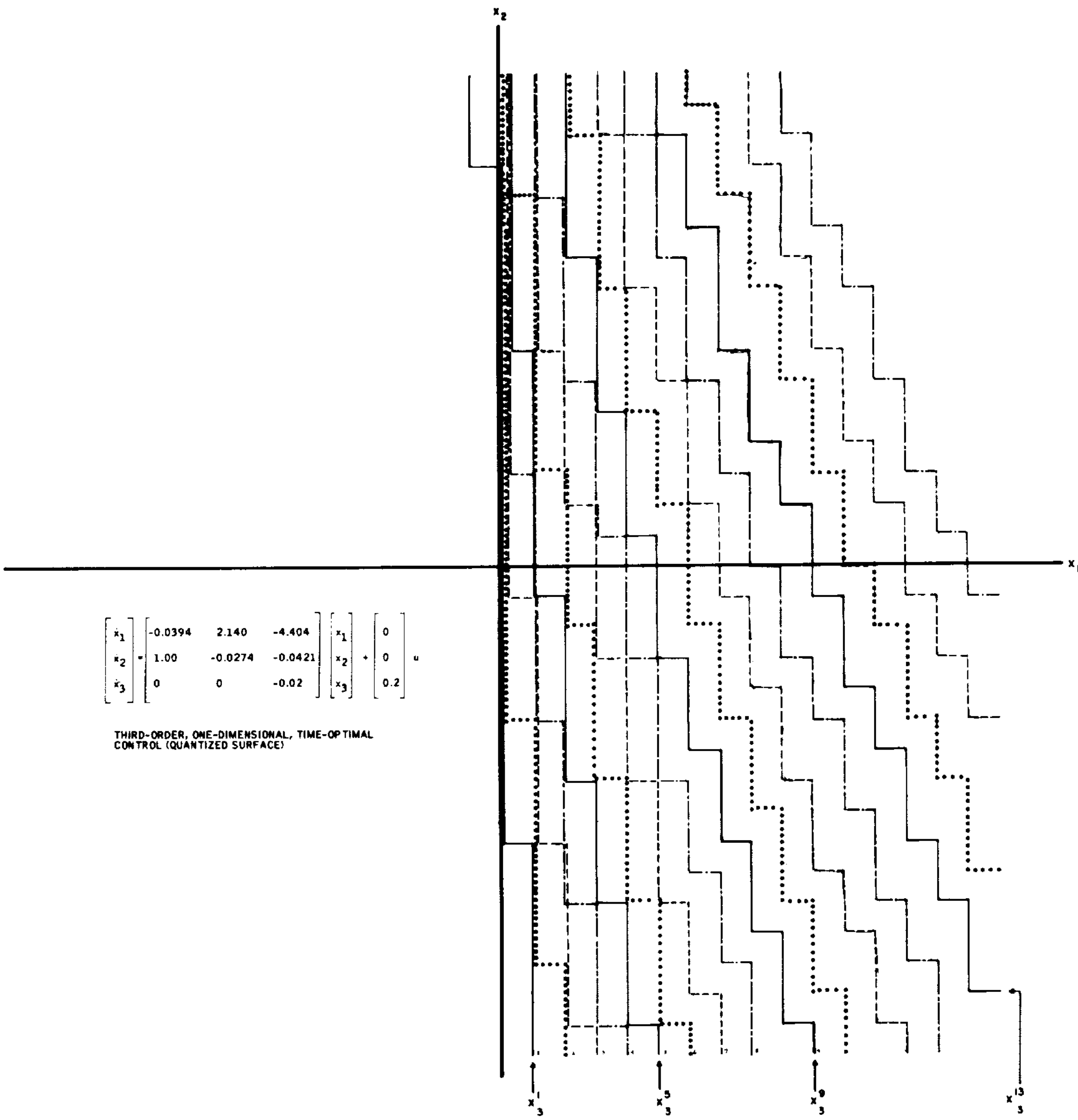


Figure 3. - Quantized third-order switching surface

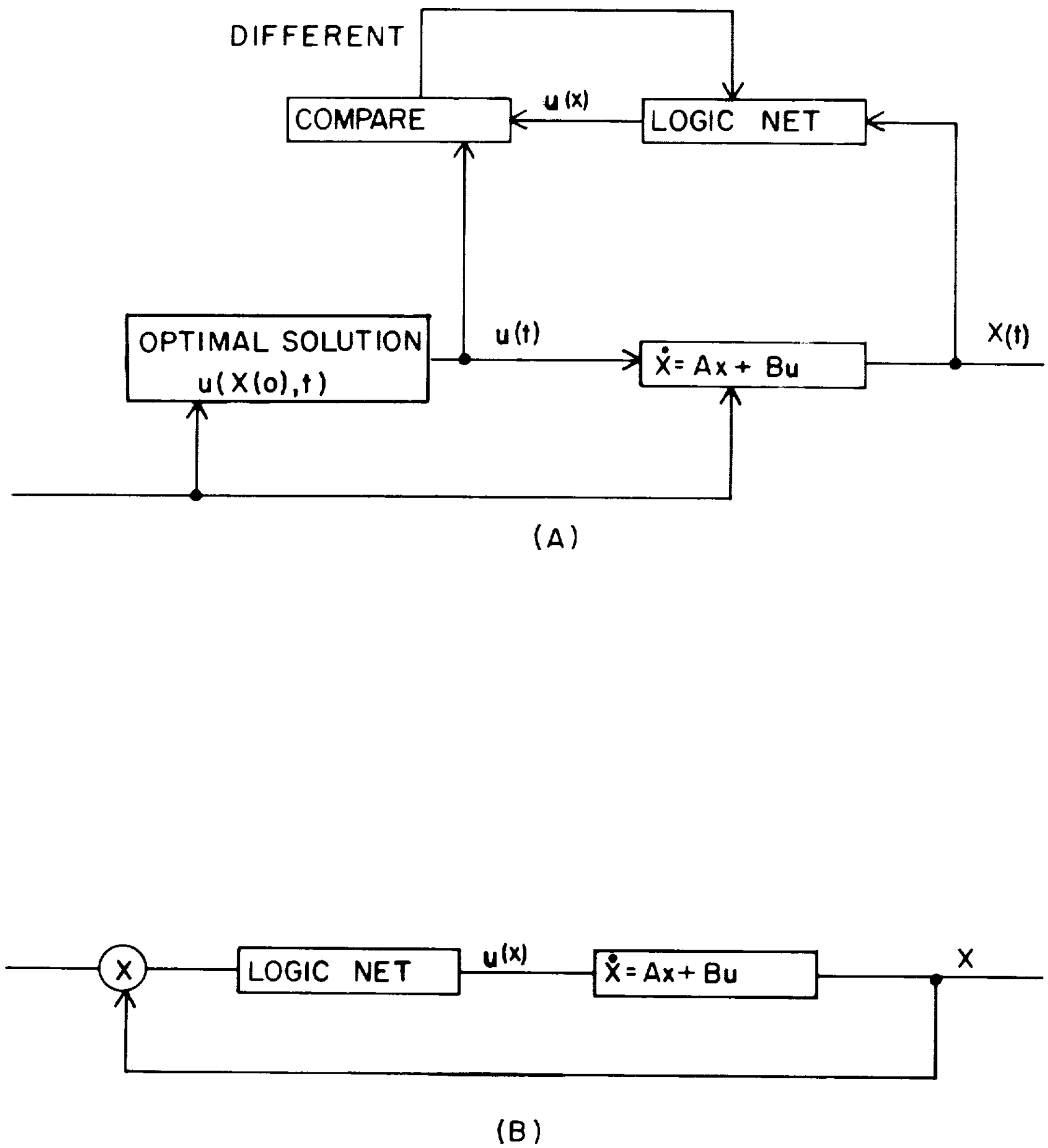


Figure 4. - Training and use of the logic net

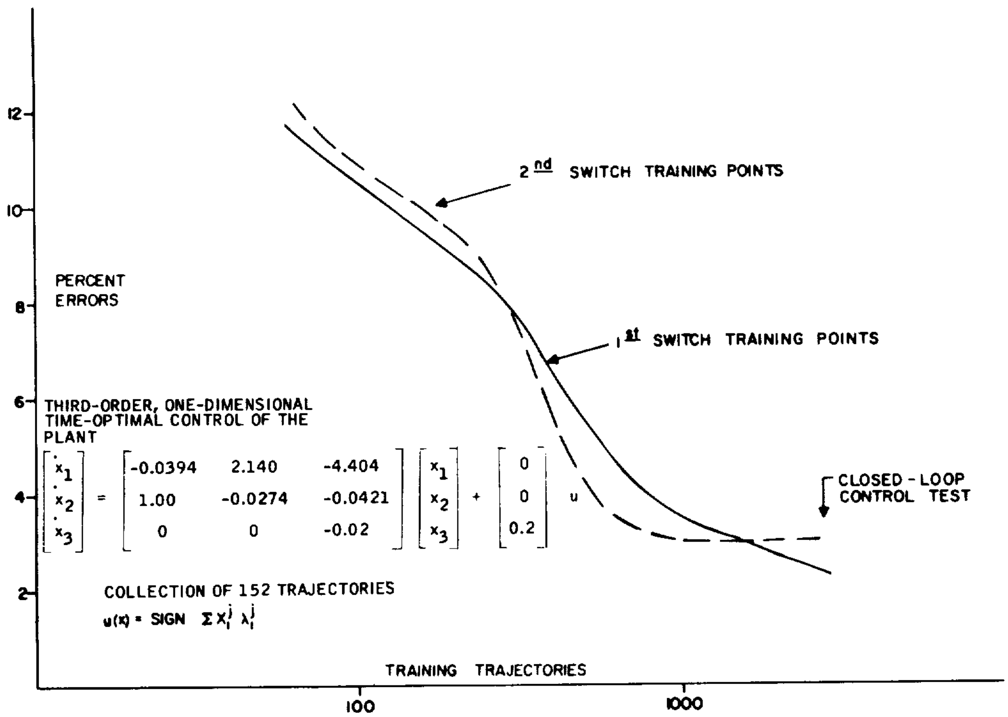


Figure 5. - Training curve (third-order)

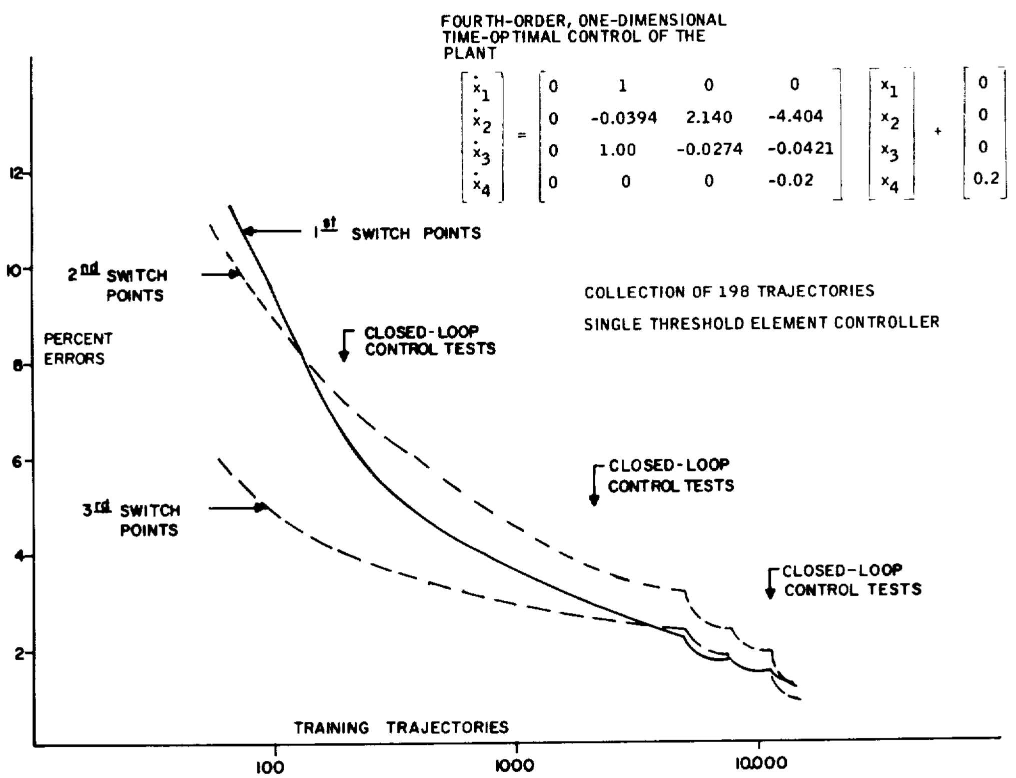
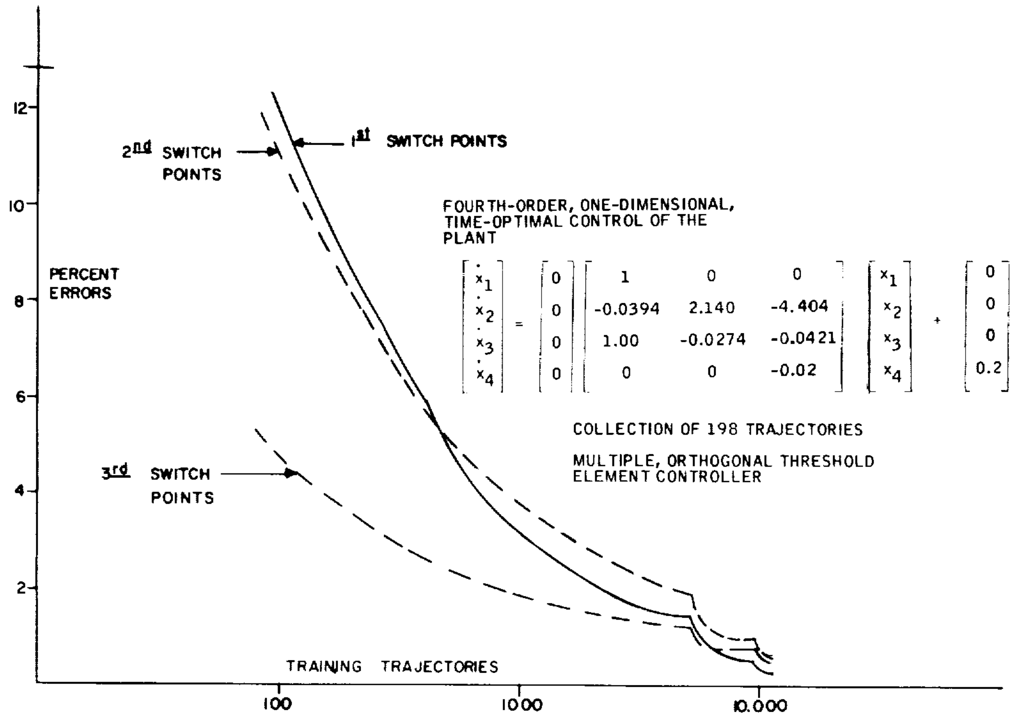


Figure 6. - Training curve (fourth-order)



$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -0.0394 & 2.140 & -4.404 \\ 1.00 & -0.0274 & -0.0421 \\ 0 & 0 & -0.02 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.2 \end{bmatrix} u$$

Figure 7. - Training curve (fourth-order)

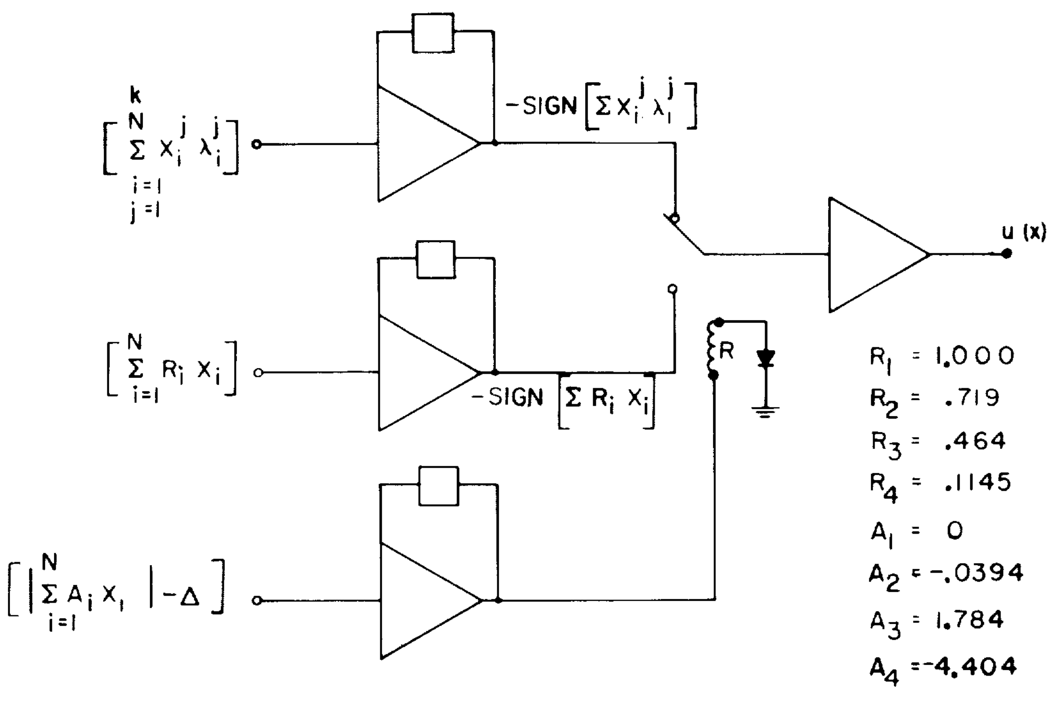


Figure 8. - Three-state control variable

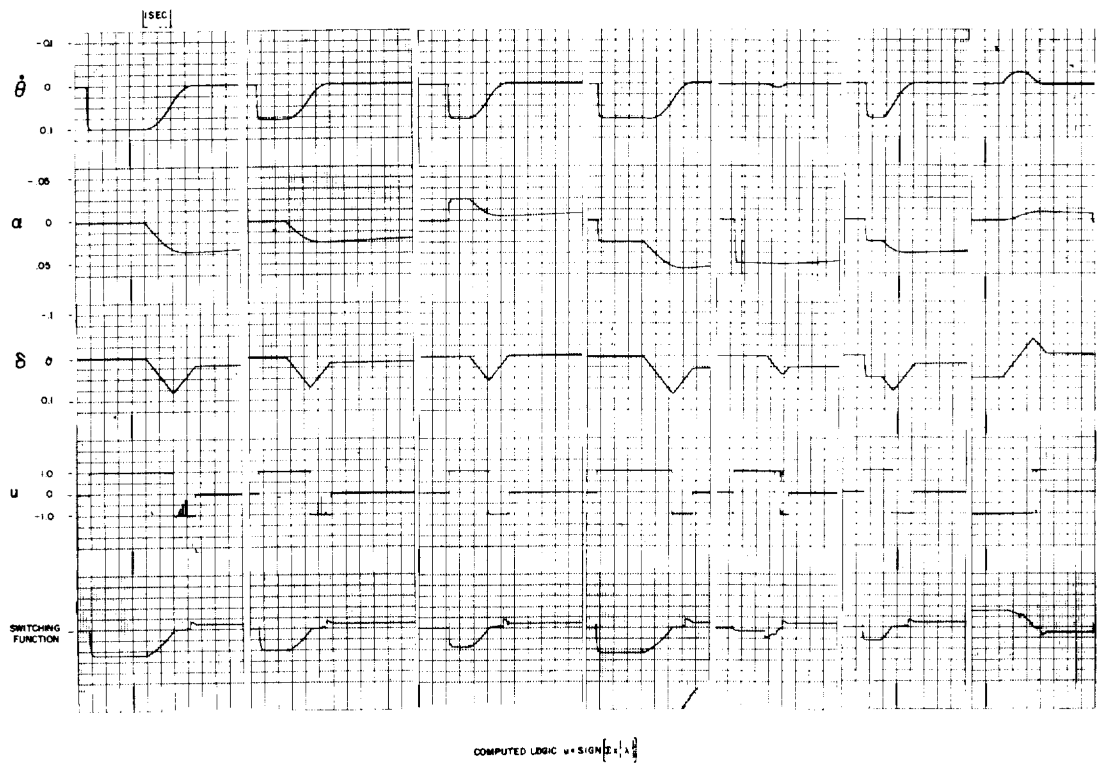


Figure 9. - Third-order one-dimensional time-optimal responses

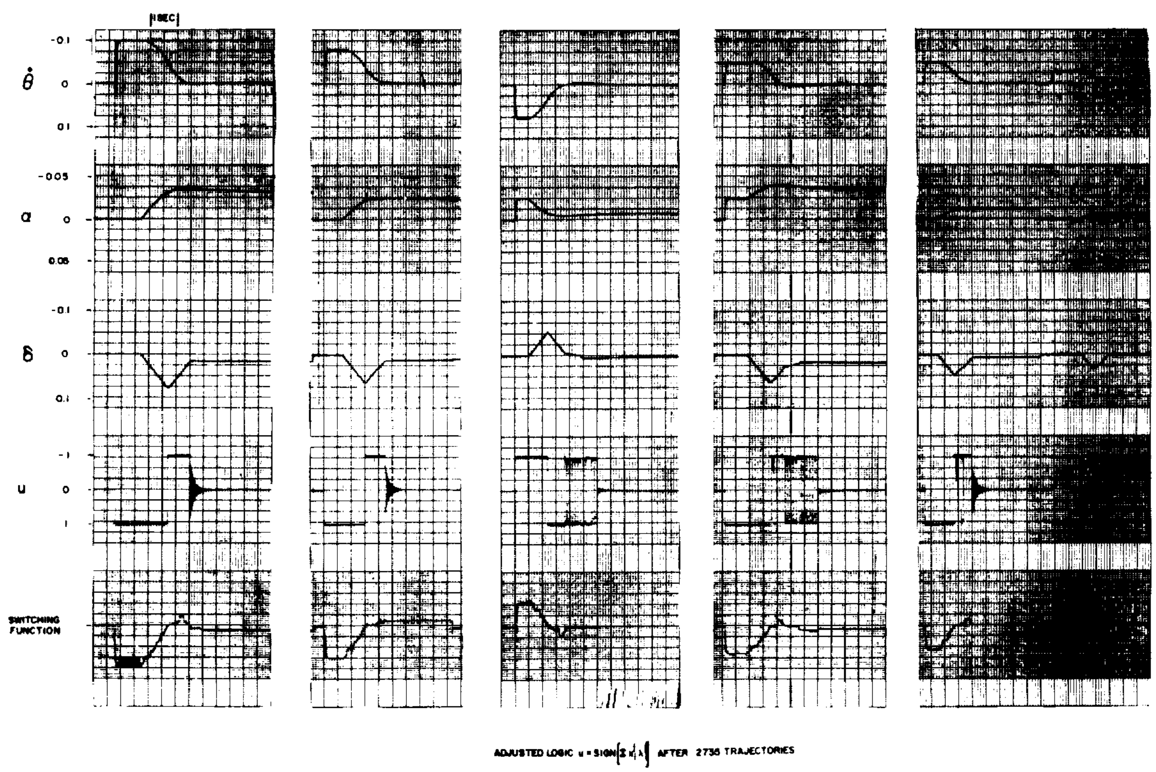


Figure 10. - Third-order one-dimensional time-optimal responses

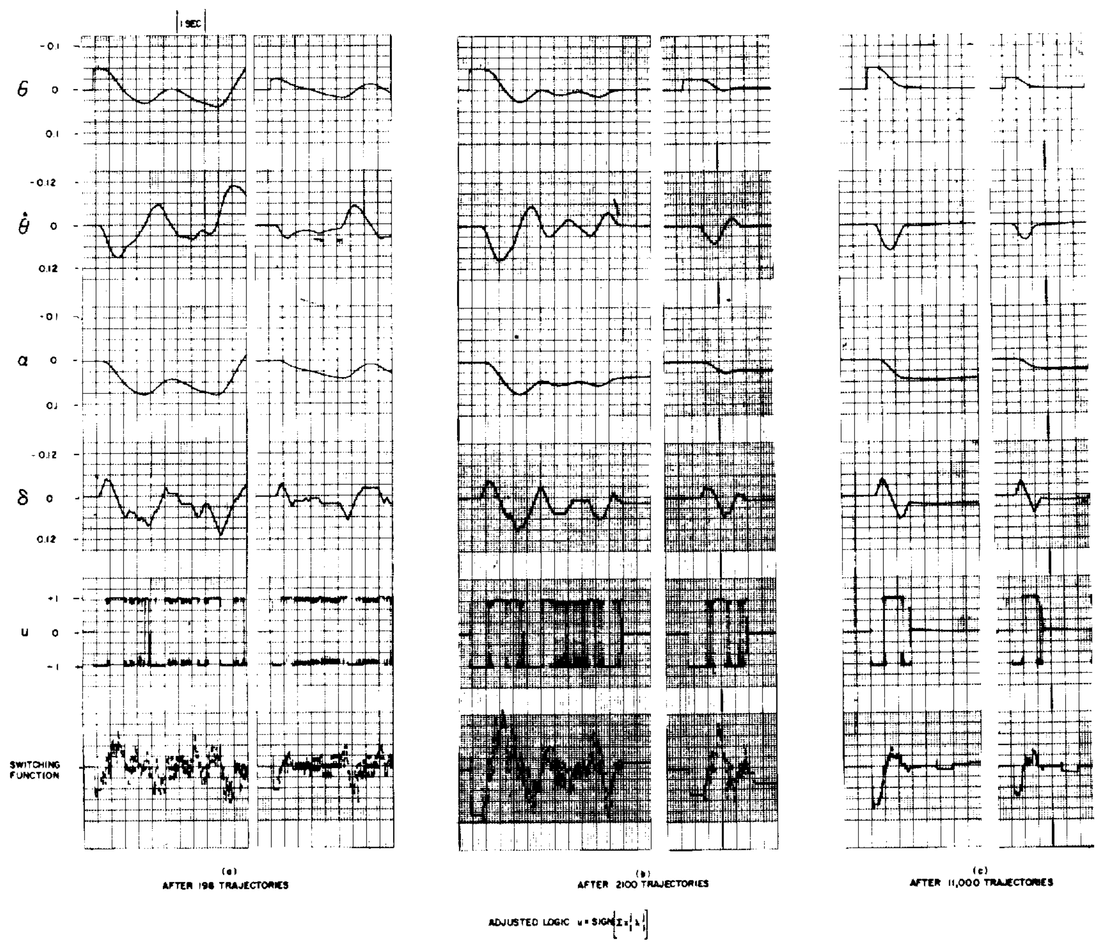


Figure 11. - Closed-loop responses at three stages of training

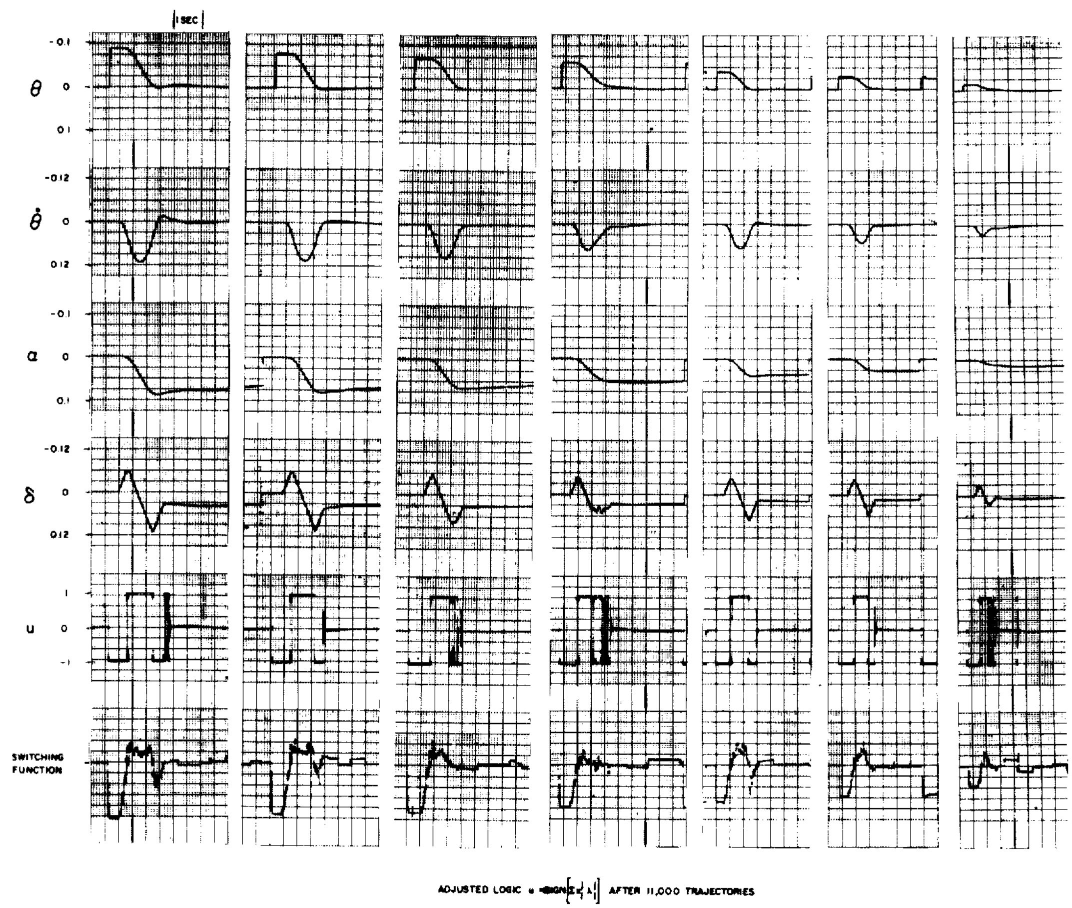
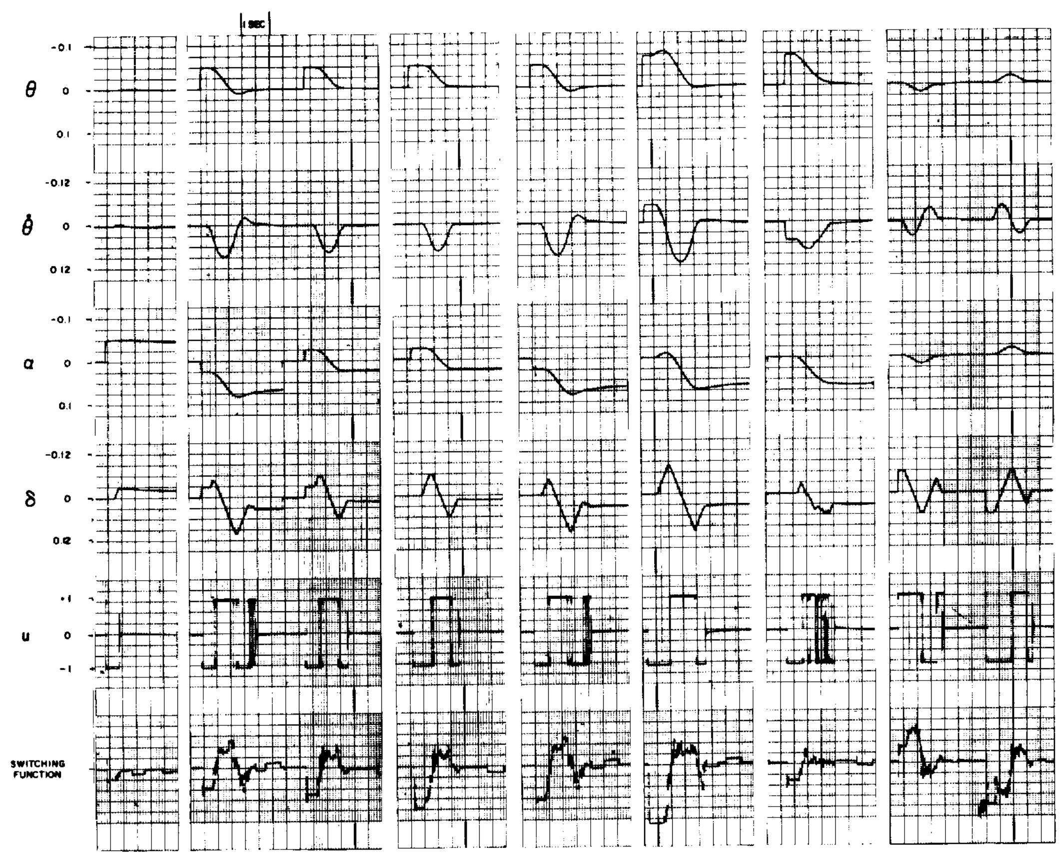


Figure 12. - Fourth-order logical net control responses



ADJUSTED LOGIC $u = \mathrm{SIGN}\left[\sum \epsilon_i^j \lambda_j\right]$ AFTER 11,000 TRAJECTORIES

Figure 13. - Fourth-order logical net control responses

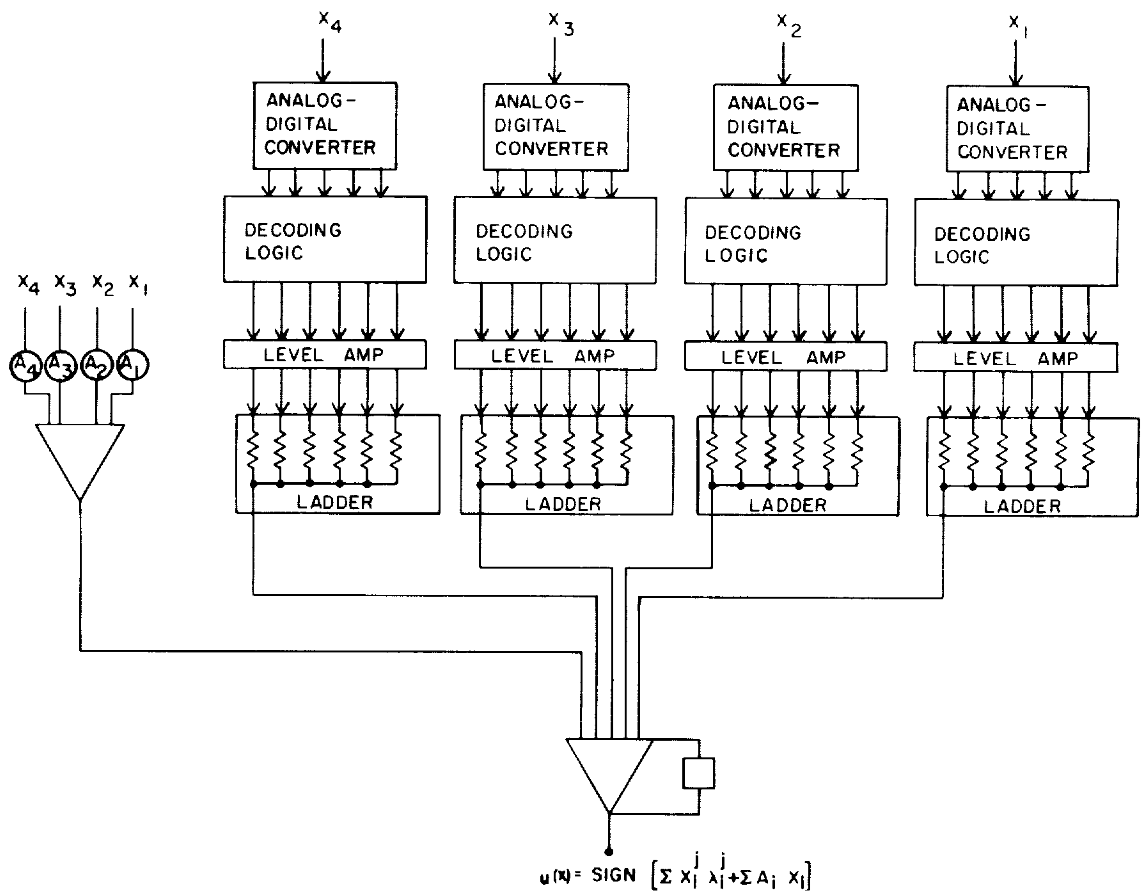


Figure 14.- Fourth-order logical net mechanization

NASA-Langley, 1962 D-1678